UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/898,583	07/03/2001	G. Glenn Henry	CNTR:2020	2877

23669 7590 05/19/2004

HUFFMAN LAW GROUP, P.C.
1832 N. CASCADE AVE.
COLORADO SPRINGS, CO 80907-7449

EXAMINER
GERSTL, SHANE F

ART UNIT	PAPER NUMBER
2183	5

DATE MAILED: 05/19/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/898,583	**Applicant(s)** HENRY ET AL.	
	Examiner Shane F Gerstl	**Art Unit** 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *2/21/02, 6/5/02, and 11/18/02*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-46* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-46* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *03 July 2001* is/are: a)□ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 3.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. Claims 1-46 have been examined.

Papers Received

2. Receipt is acknowledged of request to rescind non-publication, information disclosure statement, and change of address papers submitted, where the papers have been placed of record in the file.

Specification

3. The disclosure is objected to because of the following informalities: Pages 1, 15, 19, and 20 have missing serial numbers for copending applications.

Appropriate correction is required.

Drawings

4. The drawings are objected to because the reference characters 100 (in figures 1 and 2) and 108 (in figure 3) can be taken to point to the same element as reference characters 114 and 302. The examiner realizes that these characters are meant to refer to all the elements in the figure as a whole and recommends using a rectangular box around figures 1-3 and making the reference characters in question point to the box so as to avoid any confusion. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 38-46 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

7. Claim 38 recites the limitation "the instruction cache" in the first limitation after the transitional phrase. There is insufficient antecedent basis for this limitation in the claim. The examiner is taking the claim to mean "an instruction cache."

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1-15, 18-19, 21-24, 26-32, and 38-46 are rejected under 35 U.S.C. 102(b) as being anticipated by Emma (5,353,421).

10. In regard to claim 1,

a. an instruction cache, for outputting a line of instruction bytes selected by a fetch address (figure 10, element 13, fetch address on wire 28);

b. an instruction buffer, coupled to said instruction cache, for buffering said line of instruction bytes (figure 10, element 11);

c. a branch target address cache (figure 10, element 12, BHT and figure 11, elements 101 and 102), coupled to said fetch address (figure 10, element 28), for providing offset information relating to a location of a branch instruction within said line of instruction bytes; Figure 12 shows that the BHT (branch history table) holds branch target addresses. The included IEEE standard definition of cache simply shows that a cache is a small portion of memory used for temporary storage of frequently-used data and thus the BHT is a branch target address cache (BTAC). Further, throughout the disclosure and in figure 11, the use of the terminology "hit," which coincides with cache terminology, is used for the BHT. Column 15, lines 4-18 show that the instruction fetch address in figure 11 is for and part of a branch history table (BTAC) hit and is broken up into two parts. Part 104 is a sub-address or offset that identifies the appropriate instructions within an instruction-fetch segment.

d. and selection logic, coupled to said BTAC, for causing a portion of said instruction bytes not to be provided to said instruction buffer, based on said offset information. Figure 11 shows selection logic (elements 105, 106, and the "construct" block) for choosing the appropriate instruction address bytes to next fetch from (as seen in figure 10) for sending the line of instructions to the buffer from the cache as shown in figure 10 and detailed in column 15, lines 4-45, effectively causing the other bytes to not be provided.

11. In regard to claim 2, Emma discloses the apparatus of claim 1, wherein said offset information specifies a location of an instruction immediately following said branch instruction within said line of instruction bytes. As shown above the offset specifies the location of a branch instruction also the location of the immediately following instruction in execution order, the target instruction.

12. In regard to claim 3, Emma discloses the apparatus of claim 2, wherein said portion of said instruction bytes not provided to said instruction buffer comprises instruction bytes immediately following said branch instruction within said line of instruction bytes as specified by said offset. Column 15, lines 35-45 show that an not taken target is saved on a stack but never taken and is a "ghost hit." Since this address is never taken, the instruction buffer will not receive this instruction for execution and the instruction is prevented from being selected.

13. In regard to claim 4, Emma discloses the apparatus of claim 1, wherein said selection logic comprises: a register coupled between said instruction cache and said instruction buffer, for storing said line of instruction bytes. Figure 11, element 82 is a register within element 12 of figure 10 that is coupled to both the cache and the instruction buffer and thus coupled between them. This register has been shown in the sections cited above and in figure 11 to hold information on the line of instruction bytes.

14. In regard to claim 5, Emma discloses the apparatus of claim 4, wherein said selection logic further comprises: a plurality of valid bits coupled to said register, wherein each of said plurality of valid bits is associated with of said instruction bytes in said register. As shown in figure 11, each set of bytes includes a V or valid bit as described in column 12, lines 53-58. Column 13, lines 25-28 show that the valid bit is used to denote which of BA/TA (branch/target) pairs in each block are valid.

15. In regard to claim 6, Emma discloses the apparatus of claim 5, wherein said selection logic populates said plurality of valid bits based on said offset information received from said BTAC. It is inherent that the valid bits are populated using the offset

information since the offset information selects certain entries of the line for execution and when these entries are placed back in the BTAC they are denoted as valid or not, depending on a hit or miss, only because they had been selected.

16. In regard to claim 7, Emma discloses the apparatus of claim 6, wherein said selection logic causes each of said instruction bytes in said register having a corresponding valid bit that indicates said corresponding instruction byte is invalid to not be provided to said instruction buffer. Column 15, lines 35-45 show that an invalid target is saved on a stack but never taken and is a "ghost hit." Since this address is never taken, the instruction buffer will not receive this instruction for execution and the instruction is prevented from being selected.

17. In regard to claim 8, Emma discloses the apparatus of claim 7, wherein said BTAC provides a hit signal to said selection logic for indicating whether or not said fetch address hit in said BTAC. As shown in figure 11, the BTAC also provides a hit signal, showing that a match was found in the BTAC (BHT), to the "construct" block of the selection logic.

18. In regard to claim 9, Emma discloses the apparatus of claim 8, wherein said selection logic populates said plurality of valid bits based on said offset information received from said BTAC if said hit signal indicates said fetch address hit in said BTAC. As shown above, the valid bits are set for the selected entries and inherently the only way to tell if the branch/target information was correct is to see if there was a hit in the cache, if not, other data must be used.

19. In regard to claim 10, Emma discloses the apparatus of claim 5, wherein said selection logic further comprises: muxing logic, coupled between said instruction cache and said instruction buffer, for causing ones of said instruction bytes indicated as valid by said associated valid bit to be provided to said instruction buffer. Since muxing logic is known in the art to be simply selection logic, elements 105 and 106 and the "construct" block (2 input/1 output = multiplexer) are the multiplexing logic for sending instruction bytes to the instruction buffer from a cache holding instruction bytes.

20. In regard to claim 11, Emma discloses the apparatus of claim 10, wherein said muxing logic comprises a set of muxes for discarding ones of said instruction bytes indicated as invalid by said associated valid bit. Muxing logic inherently comprises muxes (multiplexers) or selectors for selection and as shown above the selection logic prevents invalid instruction bytes (targets) from reaching the instruction buffer.

21. In regard to claim 12, Emma discloses the apparatus of claim 10, wherein said muxing logic comprises a set of muxes for aligning ones of said instruction bytes indicated as valid by said associated valid bit with a first empty location in said instruction buffer. The construction mux of figure 11 aligns instruction bytes as described in column 15, lines 35-45 and since instructions are sent to the buffer it is inherent that an empty position is occupied, which may be called the "first" location.

22. In regard to claim 13, Emma discloses the apparatus of claim 10, wherein said muxing logic comprises a set of muxes for shifting ones of said instruction bytes indicated as valid by said associated valid bit by a number of bytes shifted out of said instruction buffer. It is inherent that if a byte of the instruction buffer is shifted out and a

new instruction byte is ready that the muxes will shift that new instruction byte into the buffer.

23. In regard to claim 14, Emma discloses the apparatus of claim 13, wherein said selection logic is configured to receive a shift count from instruction format logic for indicating a number of instruction bytes to be shifted out of said instruction buffer. The instruction format logic is the decoder of figure 10, since it identifies the format of instructions, and the next instruction register, which prepares the next instruction for decoding. Since the selection logic sends or shifts in instruction bytes to the buffer, it inherently must know when an entry is free from an entry shifted out. This notification of a shifted instruction or shift count signal is sent by the format logic, which took the instruction information from the buffer.

24. In regard to claim 15, Emma discloses the apparatus of claim 14, wherein said muxing logic shifts said ones of said instruction bytes indicated as valid by said associated valid bit by said shift count. Since the muxing logic receives indication of free instruction bytes, the muxing logic selects appropriate instruction byte for shifting into the buffer (which only holds valid bytes).

25. In regard to claim 18, Emma discloses the apparatus of claim 1, wherein said instruction buffer is directly coupled to instruction format logic that formats said instruction bytes. The instruction format logic is the decoder of figure 10, since it identifies the format of instructions, and the next instruction register, which prepares the next instruction for decoding. Figure 10 shows that the instruction buffer is directly coupled to these elements.

26. In regard to claim 19, Emma discloses the apparatus of claim 18, wherein a bottom byte of said instruction buffer is provided directly to a portion of said instruction format logic configured to a first byte of an instruction for formatting. As shown above the format logic receives an instruction provided by the instruction buffer. The byte that holds the instruction for formatting may be appropriately named the "bottom" byte and the instruction may be named the "first" instruction for formatting.

27. In regard to claim 21, Emma discloses the apparatus of claim 1, wherein said BTAC is configured to provide a target address of said branch instruction in response to said fetch address (figure 12, element 32).

28. In regard to claim 22, Emma discloses the apparatus of claim 21, wherein said target address is selectively provided to said instruction cache as a subsequent fetch address for selecting a second line of instruction bytes containing a target instruction of said branch instruction in said instruction cache. Figure 11 shows that target addresses are selectively provided and figure 10 shows that these addresses are provided to the for fetching the next line from the cache as the next instruction address on line 28.

29. In regard to claim 23, Emma discloses the apparatus of claim 22, wherein said selection logic cause said target instruction to be provided to said instruction buffer adjacent to said branch instruction within said instruction buffer. Figure 11 and column 15, lines 35-45 show that show that the branch and target address are paired and adjacent went sent for fetching and thus when fetched and placed in the buffer, they are adjacent to one another.

30. In regard to claim 24, Emma discloses the apparatus of claim 23, wherein said selection logic causes instruction bytes preceding said target instruction in said second line to be discarded and not provided to said instruction buffer. As column 15, lines 15-45 show, a branch instruction is selected along with a target and all other instructions are prevented including the immediately sequential one.

31. In regard to claim 26, Emma discloses a pre-decode stage (all blocks described below are before decode in figure 10 and thus pre-decode) within a microprocessor (figure 10), comprising:

a. an instruction buffer (figure 10, element 11), for buffer instruction data for provision to instruction format logic (figure 10, elements 16 and 17); The decoder are next instruction register are instruction format logic because the register holds the instruction to be decoded and the decoder identifies the format of the instruction.

b. selection logic, coupled to said instruction buffer, for receiving first instruction data selected by a fetch address from an instruction cache, said first instruction data including a branch instruction; Figure 11 shows selection logic (elements 105, 106, and the "construct" block) for choosing the appropriate instruction data to next fetch from (as seen in figure 10) based on data received from the cache, that is the current instruction line, where at least one of those instructions is a branch (column 15, lines 2-45).

c. and a branch target address cache (BTAC) (figure 10, element 12, BHT), coupled to said selection logic, for providing a target address of said branch instruction as a next fetch address to said instruction cache; Figure 12 shows that the BHT (branch history table) holds branch target addresses. The included IEEE standard definition of cache simply shows that a cache is a small portion of memory used for temporary storage of frequently-used data and thus the BHT is a branch target address cache (BTAC). Further, throughout the disclosure and in figure 11, the use of the terminology "hit," which coincides with cache terminology, is used for the BHT. Figure 10 shows that this unit provides the next instruction address to the cache for fetching and figure 12 shows this to be a BA/TA pair or branch address/target address pair.

d. wherein said selection logic is configured to receive second instruction data selected by said target address from said instruction cache, said second instruction data including a target instruction of said branch instruction; When there is a branch in the next line of instructions fetched from the cache, the second data selected by the target address will be received by the selection logic and as shown in figure 11 and column 15, lines 35-45 there is a target instruction.

e. and wherein said selection logic is configured to write said branch instruction and said target instruction immediately adjacent to one another into said instruction buffer. Figure 11 and column 15, lines 35-45 show that show that the branch and target address are paired and adjacent went sent for fetching and thus when fetched and placed in the buffer, they are adjacent to one another.

32. In regard to claim 27, Emma discloses the pre-decode stage of claim 26, wherein said BTAC is configured to provide said target address in response to said fetch address (column 15, lines 35-45 and figure 11).

33. In regard to claim 28, Emma discloses the pre-decode stage of claim 26, wherein said BTAC is configured to provide to said selection logic an indication of a location in said first instruction data that immediately follows said branch instruction. As shown above, the branch is paired with a target, which immediately follows the branch instruction in execution order on a taken branch and figures 11 and 12 show that the targets are provided by the BTAC. If the target is to be taken, the T-bit indicates so as shown in figure 11 and as described in column 15

34. In regard to claim 29, Emma discloses the pre-decode stage of claim 28, wherein said selection logic writes said branch instruction and said target instruction immediately adjacent to one another based on said indication of said location. Since the indication is the location of the target, this is the basis of placing the instructions adjacent in the buffer when there is a taken branch (indicated by the T bit in figure 11 and as described in column 15).

35. In regard to claim 30, Emma discloses the pre-decode stage of claim 29, wherein said selection logic is configured to receive said target address as shown above and in figure 11.

36. In regard to claim 31, Emma discloses the pre-decode stage of claim 30, wherein said selection logic writes said branch instruction and said target instruction immediately

adjacent to one another based on said target address and said indication of said location as shown above.

37. In regard to claim 32, Emma discloses the pre-decode stage of claim 28, wherein said BTAC is configured to provide said indication in response to said fetch address. As shown in figure 11 and column 15, the T-bit and target address are selected from the BTAC by the fetch address.

38. In regard to claim 38, Emma discloses a method for providing a branch instruction and a target instruction to an instruction buffer (figure 10, element 11), the method comprising:

a. receiving from an instruction cache (figure 10, element 13) a first cache line containing the branch instruction;

b. receiving from a branch target address cache (BTAC) (figure 10, element 12, BHT) an offset within said first cache line of an instruction immediately following the branch instruction; Figure 12 shows that the BHT (branch history table) holds branch target addresses. The included IEEE standard definition of cache simply shows that a cache is a small portion of memory used for temporary storage of frequently-used data and thus the BHT is a branch target address cache (BTAC). Further, throughout the disclosure and in figure 11, the use of the terminology “hit,” which coincides with cache terminology, is used for the BHT. Column 15, lines 4-18 show that the instruction fetch address in figure 11 is for and part of a branch history table (BTAC) hit and is broken up into two parts. Part 104 is a sub-address or offset that identifies the appropriate instructions within an instruction-fetch segment.

c. receiving from the instruction cache a second cache line containing the target instruction, said second cache line selected by a target address of the branch instruction provided by said BTAC; Figure 11 and column 15, lines 35-45 show that a target address is selected and used for getting the target line (on line 28 of figure 10).

d. discarding instructions after the branch instruction in said first cache line; When the branch is taken and the target is next fetched (as shown in column 15), any other instructions are inherently discarded from the cache line that were in sequential order from the branch since they will not be taken.

e. discarding instructions preceding the target instruction in said cache line; It is also inherent that since the target instruction is the next to be executed, any instructions before this point in the cache must be discarded.

f. providing to the instruction buffer a portion of said first and second cache lines remaining after each of said discardings. Since the next two instructions to be executed are the branch and target instructions, these instructions, which are portions of the cache lines, are sent to the instruction buffer.

39. In regard to claim 39, Emma discloses the method of claim 38, wherein said discarding instructions after the branch instruction in said first cache line is performed based on said offset. As shown in column 15, lines 2-45, the second part of the fetch

address is an offset that locates the appropriate instruction where the others are discarded.

40. In regard to claim 40, Emma discloses the method of claim 39, wherein said discarding instructions preceding the target instruction in said second cache line is performed based on said target address. Since, the target address shows the next instruction fetched and executed and thus it is based on this that the instructions not to be executed because they are not the next instruction are discarded.

41. In regard to claim 41, Emma discloses the method of claim 38, further comprising:

a. providing a fetch address to the instruction cache prior to said receiving from the instruction cache said first cache line;

b. wherein the instruction cache provides said first cache line in response to said fetch address.

It is inherent that the cache must receive a fetch address before receiving a cache line from the cache. A cache line is in response to a fetch address as is shown in figure 10 where the cache receives a fetch address on line 28.

42. In regard to claim 42, Emma discloses the method of claim 41, further comprising:

a. providing said fetch address to the BTAC prior to said receiving from said BTAC said offset;

b. wherein said BTAC provides said offset in response to said fetch address.

Figure 11 and column 15 show that the offset (part 104) is provided from the fetch address and used to select a position in the line and thus the fetch address must be received so the line can be provided before the offset is received fro use. The BTAC also provides the offset as the fetch address held in figure 11 is viewed as a part of the BTAC.

43. In regard to claim 43, Emma discloses the method of claim 38, further comprising: storing said first cache line in a register prior to said discarding instructions after the branch instruction in said first cache line. Figure 11, element 82 is a register within element 12 of figure 10 that is coupled to both the cache and the instruction buffer and thus coupled between them. This register has been shown in the sections cited above and in figure 11 to hold information on the line of instruction bytes and since the instruction address are selected here, the discarding takes place here.

44. In regard to claim 44, Emma discloses the method of claim 43, wherein said discarding instructions after the branch instruction in said first cache line comprises marking said instructions after the branch instruction in said register invalid, and not providing said instructions marked invalid in said register to the instruction buffer. Column 15, lines 35-45 show that an invalid target is saved on a stack but never taken and is a "ghost hit." Since this address is never taken, the instruction buffer will not receive this instruction for execution and the instruction is prevented from being selected.

45. In regard to claim 45, Emma discloses the method of claim 38, further comprising: storing said second cache line in a register prior to said discarding

instructions preceding the target instruction in said second cache line. Figure 11, element 82 is a register within element 12 of figure 10 that is coupled to both the cache and the instruction buffer and thus coupled between them. This register has been shown in the sections cited above and in figure 11 to hold information on the line of instruction bytes and since the instruction address are selected here, the discarding takes place here.

46. In regard to claim 46, Emma discloses the method of claim 45, wherein said discarding instructions after the branch instruction in said first cache line comprises marking said instructions after the branch instruction in said register invalid, and not providing said instructions marked invalid in said register to the instruction buffer. Column 15, lines 35-45 show that an invalid target is saved on a stack but never taken and is a "ghost hit." Since this address is never taken, the instruction buffer will not receive this instruction for execution and the instruction is prevented from being selected.

Claim Rejections - 35 USC § 103

47. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

48. Claims 16-17 and 33-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Emma in view of Schwendiger (6,250,821).

49. In regard to claim 16,

a. Emma discloses the apparatus of claim 1,

b. Emma does not disclose wherein said instruction buffer comprises a shift register.

c. Schwendiger discloses an apparatus for processing branch instructions that includes an instruction buffer that comprises a shift register to store instructions before sent to a decoder as shown in the abstract.

d. Column 1, line 66 – column 2, line 14 show that the invention of Schwendiger allows for more versatility and is able to accept more instructions from memory at a time resulting in speedup compared to the prior art. This versatility and speedup would have motivated one of ordinary skill in the art to modify the design of Emma to include the instruction buffer of Schwendiger that uses a shift register.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Emma to use the instruction buffer disclosed by Schwendiger that embodies a shift register so that greater versatility and system speed is realized.

50. In regard to claim 17, Emma in view of Schwendiger discloses the apparatus of claim 16, wherein said shift register is a byte-wide. Column 4, lines 61-63 show that four 32-bit (4-byte) instructions are sent to lines of the shift register instruction buffer and thus the shift register is at least a byte wide.

51. In regard to claim 33,

a. Emma discloses the pre-decode stage of claim 26,

b. Emma does not disclose wherein said instruction buffer comprises a shift register.

c. Schwendiger discloses an apparatus for processing branch instructions that includes an instruction buffer that comprises a shift register to store instructions before sent to a decoder as shown in the abstract.

d. Column 1, line 66 – column 2, line 14 show that the invention of Schwendiger allows for more versatility and is able to accept more instructions from memory at a time resulting in speedup compared to the prior art. This versatility and speedup would have motivated one of ordinary skill in the art to modify the design of Emma to include the instruction buffer of Schwendiger that uses a shift register.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Emma to use the instruction buffer disclosed by Schwendiger that embodies a shift register so that greater versatility and system speed is realized.

52. In regard to claim 34, Emma in view of Schwendiger discloses the apparatus of claim 33, wherein said shift register is a byte-wide. Column 4, lines 61-63 show that four 32-bit (4-byte) instructions are sent to lines of the shift register instruction buffer and thus the shift register is at least a byte wide.

53. In regard to claim 35,

a. Emma in view of Schwendiger discloses the pre-decode stage of claim 33,

b. Emma in view of Schwendiger does not explicitly disclose wherein said selection logic writes said branch instruction and said target instruction immediately adjacent to a last valid data byte in said instruction buffer.

c. The examiner is taking official notice that a first-in first-out (FIFO) buffer is well and known in the art and it would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Emma to use a FIFO buffer as the instruction buffer disclosed therein so that a simple buffer model is utilized. With this FIFO buffer in place it would be inherent that the branch and target instruction are adjacent to the last valid byte in the buffer since that is where the next open space would be.

54. In regard to claim 36, Emma in view of Schwendiger discloses the pre-decode stage of claim 33, wherein said selection logic writes said branch instruction and said target instruction to a next empty location in said instruction buffer. It is inherent that the instructions are written to an empty location in the buffer and the term "next" is simply a name since no relative position of what the instructions are next to is given.

55. In regard to claim 37, Emma in view of Schwendiger discloses the pre-decode stage of claim 33, wherein said instruction buffer is directly coupled to said instruction format logic. Figure 10 shows that the instruction buffer (element 11) is in fact directly coupled to the instruction format logic (elements 16 and 17).

56. Claims 20 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Emma in view of Miller (6,081,884).

57. In regard to claim 20,

a. Emma discloses the apparatus of claim 1,

b. Emma does not discloses explicitly wherein said branch instruction comprises an x86 branch instruction.

c. Miller had disclosed in the abstract the use of variable length instructions for processing.

d. Miller has shown in column 1, lines 14-16 that variable length instructions allow for efficient utilization of storage space and memory. Lines 16-17 show that the x86 instruction set is variable length. This efficient utilization of memory would have motivated one of ordinary skill in the art to modify the design of Emma to use the variable length x86 instruction set (including x86 branches) as taught by Miller.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Emma to use the variable length x86 instruction set including x86 branches as taught by Miller so that efficient utilization of memory space is realized.

58. In regard to claim 25,

a. Emma discloses the apparatus of claim 1,

b. Emma does not disclose wherein said instruction cache stores variable length instructions for execution by the microprocessor.

c. Miller had disclosed in the abstract the use of variable length instructions for processing.

d. Miller has shown in column 1, lines 14-16 that variable length instructions allow for efficient utilization of storage space and memory. This efficient utilization of memory would have motivated one of ordinary skill in the art to modify the design of Emma to use variable length instructions as taught by Miller. Thus the instruction cache (which caches all instructions) now caches variable length instructions.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Emma to use variable length instructions as taught by Miller so that efficient utilization of memory space is realized.

Conclusion

59. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

60. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The following references have been cited to further show the art with respect to branch target address caching in general.

US Pat No 5,805,877 to Black discloses a processor with a BTAC and instruction buffer.

US Pat No 5,948,100 to Hsu shows a system for processing variable length x86 instructions with a BTAC using an instruction buffer and selection logic.

US Pat No 5,850,543 to Shiell teaches a microprocessor with a BTAC and instruction buffer with interface to cache and selection logic.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (703)305-7305. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703)305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
May 17, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100